

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1-8. Canceled.

9. (Previously presented) The cache assignment method of claim 10, the method further comprising forwarding any data requests associated with non-conflicting cachelet pointers to cachelets identified by the respective pointers.

10. (Previously presented) A cache assignment method, comprising:
receiving plural data requests, each associated with respective cachelet pointers,
determining whether any of the cachelet pointers conflict with any other cachelet pointers,
if a conflict occurs among cachelet pointers, forwarding one of the data requests associated with a conflicting cachelet pointer to the identified cachelet, and
reassigning data requests associated with remaining conflicting cachelet pointers to unused cachelets.

11. (Original) The cache assignment method of claim 10, wherein multiple data requests having a common set address are forwarded to different cachelets.

12. (Previously presented) The cache assignment method of claim 10, further comprising:
determining whether any of the cachelet pointers are valid,
forwarding data requests having valid, non-conflicting cachelet pointers to the addressed cachelet, and
assigning data requests of non-conflicting cachelet pointers to unused cachelets according to a default assignment scheme.

13. (Previously presented) The cache assignment method of claim 10, wherein copies of a single data item may be stored in multiple cachelets.

14. (Original) A cache assignment method, comprising:

receiving plural data requests and associated cachelet pointers, the cachelet pointers addressing one of a plurality of cachelets within a cache,

determining whether any of the cachelet pointers conflict with any other cachelet pointers,

forwarding non-conflicting data requests to a cachelet identified by the cachelet pointer, for the conflicting data requests, forwarding one of the conflicting data requested to the identified cachelet and

reassigning remaining conflicting data requests to unused cachelets.

15. (Previously presented) The cache assignment method of claim 14 wherein the data requests are associated with respective cachelet pointers, the method further comprising:

determining whether any of the cachelet pointers are valid, and

assigning remaining data requests to unused cachelets according to a default assignment scheme.

16. (Currently amended) A cache assignment method, comprising:

receiving plural data requests and associated cachelet pointers, the cachelet pointers addressing one of a plurality of cachelets within a cache,

determining whether any of the cachelet pointers are valid,

forwarding data requests having valid cachelet pointers to the addressed cachelet, and

reassigning ~~reassign~~ remaining data requests to unused cachelets according to a default assignment scheme.

17. (Original) The cache assignment method of claim 16, further comprising:

determining whether any of the cachelet pointers conflict with any other cachelet pointers,

forwarding any data requests associated with non-conflicting cachelet pointers to cachelets identified by the respective pointers.

18. (Original) The cache assignment method of claim 16, further comprising:

if a conflict occurs among cachelet pointers, forwarding one of the data requests associated with a conflicting cachelet pointer to the identified cachelet, and

reassigning data requests associated with remaining conflicting cachelet pointers to unused cachelets.

19-25. Canceled.

26. (Currently amended) A cache assignment method, comprising:
receiving plural data requests, wherein the data requests are associated with respective cachelet pointers,
determining whether any of the cachelet pointers are valid,
forwarding data requests having valid cachelet pointers to the addressed cachelet, and
reassigning ~~reassign~~ remaining data requests to unused cachelets according to a default assignment scheme.

27. (Previously presented) The cache assignment method of claim 26, wherein each data request includes a cachelet pointer, the method further comprising forwarding any data requests associated with non-conflicting cachelet pointers to cachelets identified by the respective pointers.

28. (Previously presented) The cache assignment method of claim 26, the method further comprising:

receiving plural data requests, each associated with respective cachelet pointers,
determining whether any of the cachelet pointers conflict with any other cachelet pointers,

if a conflict occurs among cachelet pointers, forwarding one of the data requests associated with a conflicting cachelet pointer to the identified cachelet, and

reassigning data requests associated with remaining conflicting cachelet pointers to unused cachelets.

29. Canceled.

30. (Previously presented) The cache assignment method of claim 10, further comprising forwarding the reassigned data requests in parallel with the other forwarded data requests.

31. (Previously presented) The cache assignment method of claim 14, further comprising forwarding the reassigned data requests in parallel with the non-conflicting data requests and the one conflicting data request.

32. (Previously presented) The cache assignment method of claim 16, further comprising forwarding the reassigned data requests to the unused cachelets in parallel with the other forwarded data requests.

33. (Previously presented) The cache assignment method of claim 26 further comprising forwarding the reassigned data requests in parallel with the forwarding of data requests having valid cachelet pointers.

34-36. Canceled.